

22/10/2016

السبت

د. د. د. د.

محاضرة 6

* Transport Layer

Its output are segments that are sent to network layer

- Transport Layer is on end systems

Logical process

يتصل به Layers منطوقه لبعدها

* وظيفة ال Transport layer وظيفة عمل Logical Comm. بين ال applications
أوال app. processes (يتم به ال processes الموجودة على ال app.)

* ال Net. Layer يعمل Logical Comm. بين ال end systems
* على ال app and trans. layer يتم عمل ال OS منطوقه مع Processes

* Network Layer → datagram output

* Transport Layer → segment output

ال Net. Layer هي IP (Internet protocol) كضوابط ونقاط
عليها يتم وتبعها العنوان المطلوب.

* Network Layer protocol is unreliable ("best-effort",
does not check Bandwidth, congestion, packet Loss

* Complexity is in Transport Layer instead of Net. layer,
it's up to app. developer to choose trade off between security or speed.

Trans. Layer → TCP (Transmission Control Protocol)
→ UDP (User datagram Protocol)

* App developer choose between UDP and TCP

① ~~High throughput, Low delay, no security~~

②

Check 3.4 for comparison UDP vs TCP

TCP is Slower Than UDP because of

① congestion control

② flow control

③ Error checking

TCP (reliable but slow)

UDP (unreliable but fast)

* IP في ال Net Layer من Reliable على كونه
Transport Layer و reliable و unreliable في ال Transport Layer
عنه ال reliability بتزيد بتقيد (complexity)

* ال Net. Layer بتوفر وظائف كثير سريع في ال Algorithms

* Transport Layer does Multiplexing/demultiplexing for its segments

↓
أولاً
* Mux: عندي أكثر من App قبل واحد من ال segments فأجمعها
وابعثها لل Net. Layer للتواصل

↓
ثانياً
* demux: يجمع ال segments من ال Net layer فيطبقها
وتوزعها على التطبيقات المختصة

Mux → collect and send

demux → receive and ~~data~~ distribute

* we have sockets between App and trans. layer
each socket has ID and is application specific

* when received e.g. email; segment has the
socket ID so Trans. layer knows which app to
receive the segment.

* ~~نظام~~ ~~ال~~ ~~segments~~ ~~ال~~ socket كامن في buffer، يجمع
ال segments الخاصة بكل App ويغلقها مرة واحدة.

* كل socket له ID مميز يعرفه ال App وال segments

Check 3.5 for examples

* Trans layer can do both Mux and demux
at the same time.

شكل ال segment يختلف بين UDP و TCP

headers ← $\left\{ \begin{array}{l} \text{SRC Port No.} \\ \text{dest Port No.} \end{array} \right.$

Port No. طول 16 بت، كل Source مع ال dest. 32 bit
* عند تصميم ال رقم أقدر أستخدم ال Application، فالد developer لازم يعرف
ال port اللي شغال عليه

ال Net Layer بينهما ال IP فقط ومتر بينهما بقية الداتا.

* انه اقل ما تنتقل بين ال layers كل Layer تشوف جزء من ال headers
فقط، مش كامل الرسالة، كل رسالة ينقلها جزء ال layer المناظرة
ليها.

ال SRC Port هو ال Port الی یستنی علیہ ال Client و ال Server
علی ال application .

Telnet uses UDP , HTTP uses TCP

UDP uses 8 byte header
TCP uses 20 byte header

- * UDP is connection-less , unreliable data transfer , no congestion and flow control
- * it only does checksum (data check) 16 bit by first complement of data.

- یقسم ال data ل 16 بیت

- یجمعها ویجیب ال first complement للجمع (checksum)

- ویضف ال checksum لیخزن فی ال header

- ال 16 بیت الثامنة بال checksum هي الی بینا کد بیوم ص سلامه ال data

- لا یوصل ال جمع ال segments مع ال checksum ، لو کل ال نتائج یو طایه (byte of one)
صیقة ال داتا صلیقة ، غیر کده ال داتا صر صلیقة و تقبل .

Length → to know the segment's size

segments can vary in size

أو مقله فی ال checksum لو صمد تغییر فی مکویم (two bits) صر هنظم